Study Buddy

Software Design Document

אסתר בינס 211328109
טליה מיזליש 207883430
נעה נוסבאום 206664278

תאריך: 10.1.23

תוכן עניינים:

1. מבוא

1.1 מטרה

מסמך עיצוב תוכנה - הנקרא לפעמים מפרט עיצוב תוכנה - הוא תוכנית מפורטת לפיתוח תוכנה. SDD צריך לתאר את הפונקציונליות של התוכנה המוגמרת (מפרטים) ואת התוכניות של הצוות שלך לבנות אותה (ציר זמן, יעדים וכו'). על ידי יצירת מסמך עיצוב תוכנה, צוות ההנדסה שלך ובעלי עניין אחרים יכולים לקבוע ציפיות מדויקות לפרויקט לפני שתתחיל בקידוד. למרות שאין דרך בטוחה להימנע מעיבוד מחדש של אלמנטים של הפרויקט שלך, SDD הוא מקום טוב להתחיל בו. SDD גם עוזרים לייעל את תהליך הקידוד. כדי ליצור SDD, עליך לחשוב על כל ארכיטקטורת המערכת שלך לפני כתיבת קוד כלשהו. זה מאפשר לך לצפות מראש תקלות או מחסומים ולתכנן סביבם. כשחברי צוות שונים עובדים על בניית חלקי התוכנה שלהם, ישנו מסמך מרכזי המתאר תכונות, תלות ושאר פיסות מידע שימושיות.
מטרת מסמך עיצוב תוכנה זה היא לתאר את הארכיטקטורה ואת עיצוב המערכת של הפרויקט Study Buddy, בין בהקשר של חווית המשתמש ובין בצורת אכסון הנתונים של התוכנה.

1.2 היקף

מטרות הפרויקט מתפצלות לשני תחומים עיקריים:

1. לספק מקומות עבודה ואפשרות הכנסה לסטודנטים .
2. לספק מקור למידה לסטודנטים על מנת לאפשר להם להגיע להצטיינות ולמצות את יכולותיהם .

אנחנו סטודנטיות באוניברסיטה ובמהלך הלימודים נחשפנו לפער שהפרויקט נועד לגשר עליו, כלומר הפרויקט מנצל הזדמנויות .
באוניברסיטה לומדים סטודנטים רבים, חלקם מצליחים יותר וחלקם פחות והרבה עם תחומי למידה משותפים .
מחד קיימים סטודנטים שזקוקים לסיוע בקורסים מסוימים ומאידך ישנם סטודנטים שמעוניינים להרוויח כסף מהנושא .
מטרת הפרויקט שלנו היא להפגיש בין סטודנטים אלה ולייצר פלטפורמה שמאפשרת לסטודנטים שיש להם את האפשרות להציע עזרה בתשלום, ולסטודנטים אחרים לקבל את העזרה שהם זקוקים לה בכדי להצליח בלימודיהם .

אפליקציה נוחה ויעילה למציאת שיעור פרטי, אין עוד אפליקציה שמסייעת בתחום הזה, ואפשר בפשטות ובמהירות למצוא שיעור שמתאים בצורה המירבית.
יתרון נוסף הוא שהאפליקציה מאפשרת למצוא סביבת למידה וגם זה תכונה ייחודית שמלבד הרשתות החברתיות אין עוד פלטפורמה שמוצאת פתרון טוב לבעיה הזאת.

הפרויקט שלנו מיועד לסייע לסטודנטים באוניברסיטה למצוא מורים פרטיים וקבוצות למידה מצד אחד ולספק להם הכנסה כלכלית מצד שני. האפליקציה יוצרת קשר בין סטודנטים שמחפשים ללמד סטודנטים אחרים לבין סטודנטים המחפשים מורה פרטי, וכן בין סטודנטים המחפשים להיות חלק מקבוצת למידה, כל זאת בהתאמה לצרכים של הסטודנט.
האפליקציה בנויה בעזרת התוכנה Android Studio, ונתוני האפליקציה מאוחסנים ב Firebase.

1.3 סקירה כללית

מסמך זה מסודר באופן המתואר בתוכן העניינים, ומכיל את הנושאים הבאים: מבוא, סקירת מערכת כללית, ארכיטקטורת מערכת, עיצוב רכיבים, עיצוב רכיבים, עיצוב ממשק משתמש, וטבלת דרישות. בכל פרק נתאר את הנושא הרלוונטי ונפרט עליו.

2. סקירה כללית של המערכת

בחלק זה ניתן תיאור כללי של הפונקציונליות, ההקשר והעיצוב של הפרויקט שלנו.
האפליקציה שלנו בנויה עבור שני סוגי משתמשים: מורה ותלמיד. סטודנט יכול להרשם בתור שני הסוגים, תלמיד או מורה (משתמש יכול להיות רשום פעמיים, בתור שני המשתמשים ולהיכנס לאפליקציה כל פעם בתור משתמש אחר). ההרשמה מתבצעת דרך Google.
המשימות העיקריות שמשתמש יכול לבצע בתור סטודנט הן לחפש מורה פרטי לפי פרמטרים כמו קורס מסויים, יום, טווח שעות וכו, ברשימת החיפוש הוא יכול גם למיין את רשימת המורים שמתאימים לפרמטרים שלו לפי פרמטרים נוספים כמו עד כמה הוא אהוב, עד כמה הציונים שלו טובים, עד כמה יש לו שיעור זמין בקרוב ועוד.
משימה נוספת מרכזית לסטודנט היא היכולת לבחור מורה מתוך המורים שהוצעו לו ולקבוע אצלו שיעור, כשסטודנט קובע שיעור זה מופיע בזמן אמת אצל המורה והשעה "נתפסת" (לא ניתן לקבוע שיעור נוסף אצל המורה הזה בשעה הזאת). רק משתמש בתור סטודנט יכול לקבוע שיעורים באפליקציה.
משימות מרכזיות של המורה הן העלאת שעות תאריכים בהם הוא פנוי ללמד וקורסים בהם הוא מוכן להעביר שיעורים פרטיים. בנוסף כתוב מה הציון של אותו מורה בכל קורס אותו הוא מעביר (במידה שעבר אותו בעצמו בעבר) והמחיר אותו המורה נותן לכל קורס. מורה יכול ללמד מספר שונה של קורסים וכל אחד מהם לתמחר בצורה שונה.
פעולות נוספות אפשריות באפליקציה לשני סוגי המשתמשים הם עידכון פרופיל, צפיה בשיעורים הקבועים להם בעתיד בדף הבית, ביטול שיעורים, גישה לווצאפ לצאט עם המורה/התלמיד של הקורס, שינוי סוג משתמש (מתלמיד למורה והפוך) ועוד.
בנוסף בסוף כל שיעור (כשתאריך השיעור עובר) קיים דף שמציג את השיעור והאם הוא שולם. אצל התלמיד מוצג קישור דרכו הוא יכול להיכנס ישירות לאפליקציית הפייבוקס ולשלם למורה, אצל המורה מוצגים כל השיעורים שהוא העביר ועוד לא שולמו. עבור כל אחד מהמשתמשים קיימת אפשרות לסמן האם השיעור שולם(התלמיד מסמן אחרי שהוא משלם והמורה מסמן כשהוא מקבל הודעה מפייבוקס עבור תשלום). ברגע ששני המשתמשים מסמנים שהשיעור שולם הוא נמחק מהמערכת.
עבור תלמיד קיימת אופציה נוספת של שימוש באפליקציה, בנוסף לחיפוש שיעור פרטים הסטודנט יכול לחפש סביבת למידה- קבוצה של סטודנטים איתם הוא יכול ללמוד עבור קורס מסויים, הסטודנט יוכל לחפש לפי פרמטרים מסויימים (גודל הקבוצה, זמנים, מידת רצינות וכו) קבוצה של סטודנטים שאיתם יוכל ללמוד יחד ולהתקדם.
כל המידע, כולל מידע אישי, פרטי השיעורים והקבוצות וכן הלאה נשמר בFirebase המקושר לפרויקט.
בעת קביעת שיעור, ניתן למצוא מורה פרטי על ידי סינון לפי קורס, תאריכים ועוד. וכן לגבי קבוצות למידה.
עיצוב האפליקציה נעשה בהשראת צבעי הלוגו שלה.

3. ארכיטקטורת המערכת

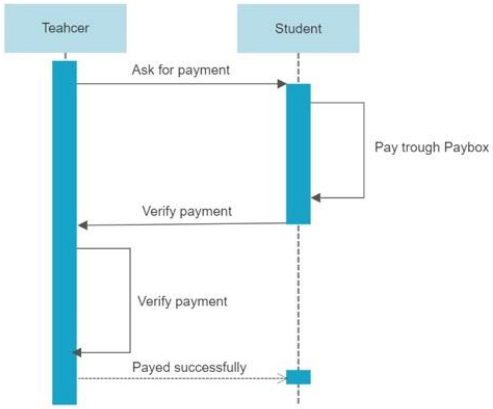
3.1 עיצוב ארכיטקטורי

בחרנו לחלק את הפרוייקט שלנו ולעבוד לפי המודל של MVVM.

- Model- תת מערכת שאחראית על הלוגיקה של הפרוייקט (פונקציות, מחלקות, אלגוריתמים, קשר למאגר הנתונים וכו)
המודל מכיל פונקציות שה- veuw model משתמש בהם וכך בא לידי ביטוי הקשר בינהם.
- veuw model -תת מערכת שאחראית על התקשורת והעברת הנתונים בין ה-model ל-veuw, הmvi מקבלת את כל הנתונים שהיא צריכה ממאגר הנתונים רק דרך המודל ושולחת אותם לveuw בשביל להציג אותם באפליקציה, ומקבלת נתונים שנקלטו מהמשתמש תוך כדי שימוש באפליקציה דרך veuw ושולחת אותם ל Model בשביל שזה יעדכן אותם במאגר הנתונים במידת הצורך.
- Veuw- תת מערכת שאחראית אך ורק על הצגת הנתונים ונראות המסכים של האפליקציה, לתת מערכת זו אין גישה לא למאגר הנתונים ולא למודלים של האפליקציה, אלא רק לveuw model דרכו היא שולחת ומקבלת את כל המידע הדרוש
- חלק נוסף וחשוב באפליקציה שהוא לא בדיוק תת מערכת הוא מאגר הנתונים שלנו ששמור בתוך פיירביס, כל משתמש חדש, בין אם מורה או תלמיד, כל שיעור שנקבע, וכל שאר הנתונים שנכנסים ונקלטים (או נמחקים) באפליקציה שמורים אצלנו בתיקיות ורשימות נפרדות בפיירביס ומתעדכנות אונליין אצל כל המשתמשים בכל מקרה של שינוי.

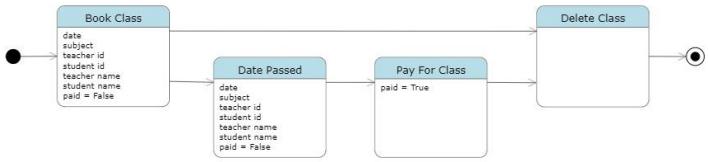
גרף המתאר את המבנה הכלל של המערכת-

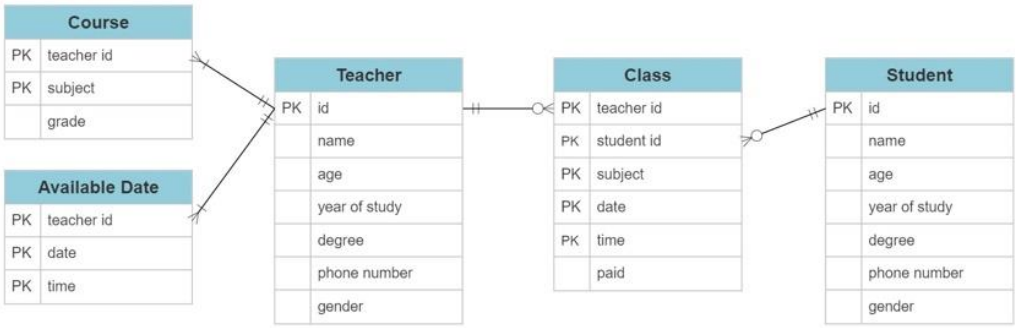
3.2 תיאור פירוק



בדיאגרמה זו מתואר הקשר בין התשלום לבין איש המשתמש, בין אם הוא מורה ובין אם הוא תלמיד. כפי שניתן לראות, התשלום נעשה דרך אפליקציית Paybox.

בדיאגרמה זו מתואר הקשר בין קביעת שיעור לבין הפונקציות המקושרות לclass זה. לאחר קביעת השיעור, נשמר התאריך בו השיעור נקבע, ניתן לשלם על השיעור ולמחוק אותו.





בדיאגרמה זו מוצג הקשר בין הclass של שיעור, המורה והתלמיד. גם מוסבר הקשר בין התאריכים הרלוונטיים עבור השיעור וכן הסינון לפי קורסים. התהליך אותו עובר התלמיד – בחירת קורס, בחירת מורה, תאריך רלוונטי, יצירת השיעור ע״י קישור התלמיד והמורה לשיעור זה, ושמירת התאריך הקורס הנבחרים.

3.3 היגיון עיצובי

בחרנו בתבנית MVVM מכיוון שהיא נוחה מאוד ומציגה היגיון ברור בעבודה עם undroid studio.
סביבת עבודה זו מובנת כך שקיימת חלוקה אוטומטית בין קבצי הxml, הם בפרוייקט שלנו מקיימים את תת המערכת שאחראית על veuw (ההצגה של הנתנים בפועל על מסך האפליקציה) לבין שאר הקוד.
בנוסף קיימות פונקציות מובנות רבות שמקלות על החיבור בין הקוד שלנו למאגר הנתונים שבחרנו לעבוד איתו(פיירביס).
בגלל שהפרוייקט שלנו מתמקד הרבה בתצוגה של נתונים, שליפה ושינוי שלהם, החלטנו שהתבנית הכי יעילה עבור הפרוייקט שלנו ותיתן לנו את התועלת המירבית היא חלוקה כזאת שמדגישה את החלקים המרכזיים של הפרוייקט, הצגה, שליפה והכנסה של נתונים, ועיבוד שלהם.
תבנית נוספת ששקלנו להשתמש בה היא תבנית השכבות. בחרנו לא להשתמש בה מכיוון שאין הרבה בדיקות לוגיקה באפליקציה שלנו ולכן העדפנו שהלוגיקה תהיה יחד עם ההתעסקות עם מאגר הנתונים, דבר שלא מתאפשר בתבנית השכבות.

נסביר את ההיגיון בבחירת המבנה שהצגנו ב3.1.
בעת פיתוח האפליקציה, דנו בנושא מבנה המשתמש. התלבטנו בין הרשמה דרך Google ולאחר מכן בחירת סוג משתמש, לבין בחירת סוג משתמש ואז הרשמה. לבסוף בחרנו בדרך השנייה, שכן הבנו שהמידע שיוחזק בכל סוג יהיה שונה, ומכאן שזו תהיה הדרך היעילה ביותר להרשמה ולכניסה לאפליקציה.
את הclass של DetailActivityTeacher בנינו בצורה הזו על מנת שנוכל לסנן שיעורים לפי מורים המלמדים את השיעור עבורו הסטודנט רוצה לקבוע שיעור.

4. עיצוב נתונים

4.1 תיאור נתונים

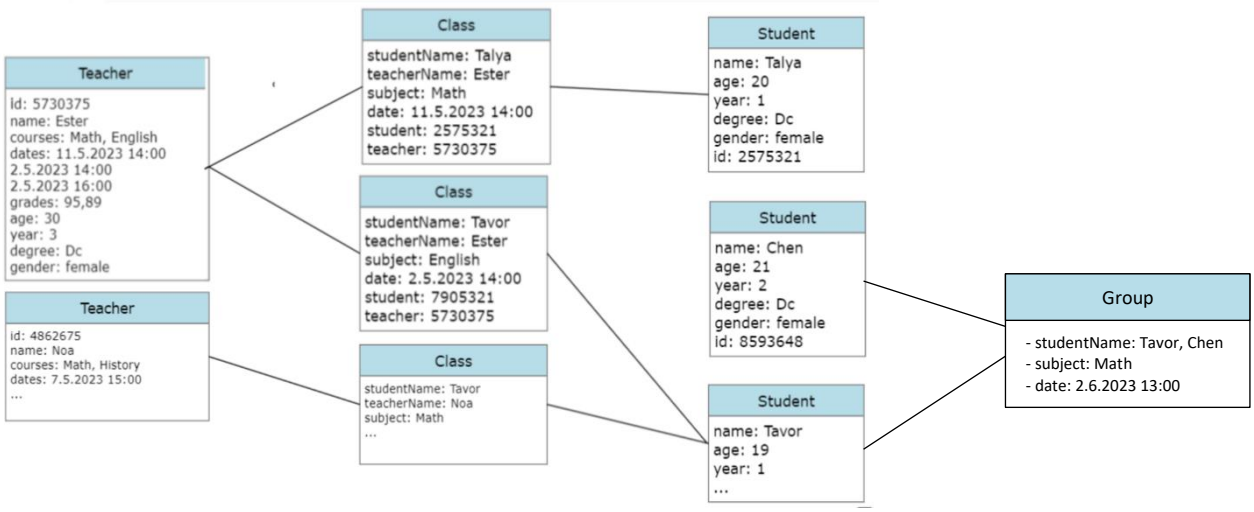
כל אובייקט מסוג מורה, תלמיד, קורס, קבוצת למידה ושיעור מוחזק בתור אובייקט מסוג Document בבסיס הנתונים Firebase שבו אנו משתמשים. בכל אחד מהם הנתונים שלו מוחזקים בתור Arraylist, כל משתנה באובייקט מכיל כותרות ותוכן המקושר לכל כותרת (שם המסביר מה המידע המוחזק שם).
לכל משתמש קיים מספר משתמש, איתו ניתן לזהות את המשתמש, ובעזרתו ניתן למצוא את השיעור ואת קבוצות הלמידה המקושרים אליו.

המידע של המערכת שלנו נשמר ב firestore, קיימים 4 collections:

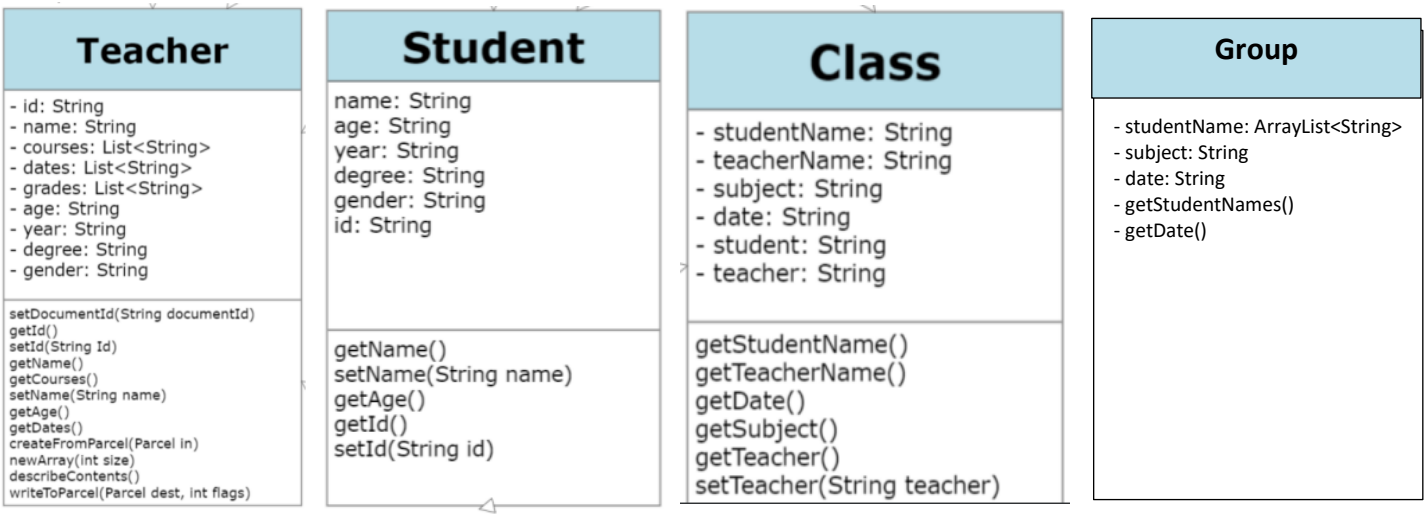
1. תלמידים
2. מורים
3. שיעור
4. קבוצה

4.2 מילון נתונים

את החלק הבא נסביר באמצעות הדיאגרמה הבאה.



בדיאגרמה זו ניתן לראות את האובייקטים classes עליהם דובר קודם, וכן את המשתנים של כל אחד. לכל אובייקט קיימות פונקציות constructor וכן פונקציות לשינוי המשתנים ושליפתן (Getters and Setter).



5. עיצוב רכיבים

על מנת לקבוע שיעור, על התלמיד לכתוב עבור איזה קורס הוא מחפש שיעור. לאחר שכתב את שם הקורס, באובייקט DetailActivityTeacher מתרחש סינון מורים לפי שמות הקורסים אותם הם מלמדים. שמות המורים הרלוונטיים נשלפים ומוצגים למסך. מכאן מועברים לעמוד BookClass, שם נשלפת רשימת הקורסים אותם מלמד המורה ולאחר בחירת הקורס יישלפו התאריכים הרלוונטיים. מכאן נותר ללחוץ על Book Class, שייצור אובייקט חדש של שיעור, בו יישמרו המספרים המזהים של המורה והתלמיד, התאריך והקורס של השיעור.
על מנת לשלם, המורה שולח לתלמיד בקשה לתשלום. התלמיד מאשר ומשלם דרך Paybox ולאחר מכן נשלח למורה אישור על מעבר תשלום. המורה מאשר את קבלת הכסף והכסף מועבר לחשבון הבנק שלו.
על מנת לקבוע קבוצת למידה עם קבוצת תלמידים, על תלמיד לבחור את הקורס בו הוא מעוניין למצוא קבוצת למידה ולאחר מכן לבחור מבין התלמידים האחרים הרלוונטיים. הוא יכול לבדוק את גילם, מינם, ואת התואר אותו הם עושים. לאחר בחירת קבוצה, התלמידים קובעים תאריך המתאים לכולם ומידע זה מעודכן במאגר הנתונים.

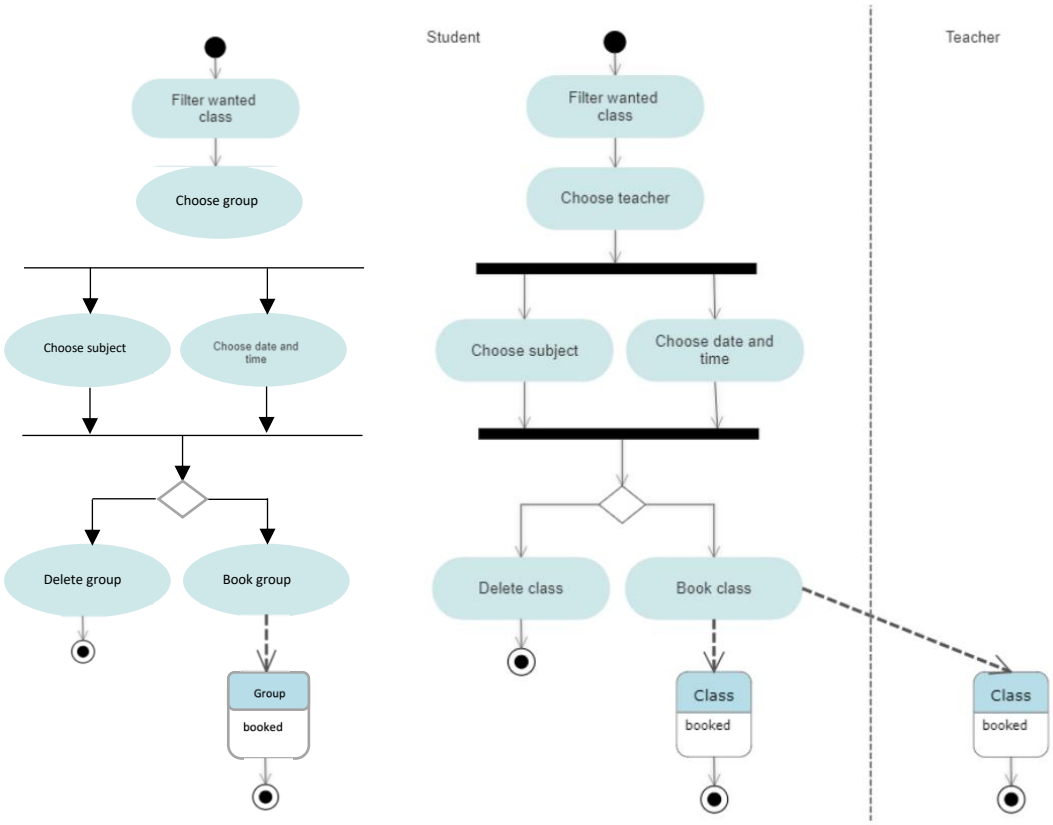
6. עיצוב ממשק משתמש

6.1 סקירה כללית של ממשק משתמש

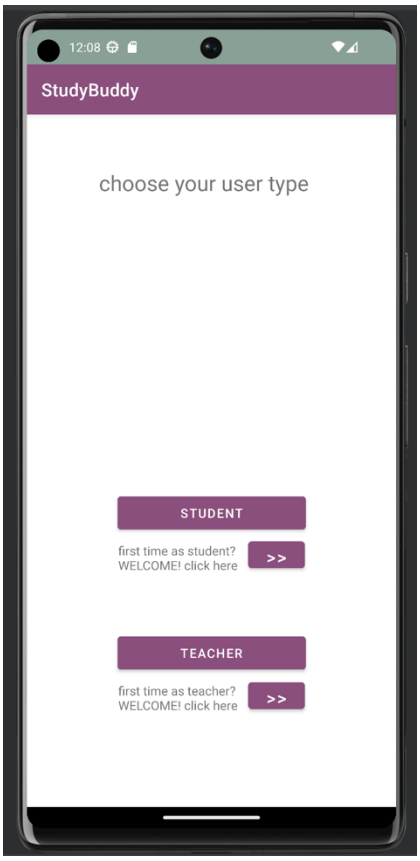
מהנקודת מבט של המשתמש- אנחנו חיפשנו פתרון ומענה לצורך שאנחנו נתקלנו בו בתור סטודנטיות, דרך קלה נוחה נגישה וברורה לתפעול למציאת מורה פרטי או סביבת למידה לפי כל הפרמטרים המסויימים שנדרשים עבורנו.
לכן האפליקציה שלנו תהיה נגישה ונוחה, בכל עמוד יש כפתור חזרה לדף הבית התפריט עם מעבר לכל אחת מהפעולות האפשריות עבור כל משתמש, כל הלחיצה על כפתור שעושה פעולה מסויימת מעבירה את המשתמש ישירות לעמוד בו הוא צריך להיות, כל דבר שנוכל לחסוך מהמשתמש מלעשות באופן עצמאי נעשה כך שהשימוש יהיה נגיש ומזמין בצורה הרבית.

תחילה, המשתמש בוחר מאיזה סוג משתמש הוא רוצה להכנס למערכת.
המשתמש יכול למלא פרטים אישיים, להחליף סוג משתמש, ולראות בעמוד הבית אילו שיעורים וקבוצות הוא קבע שעתידים להתקיים.
בתור סטודנט:
הוא יכול להכנס ל Book Class ולקבוע שיעור כפי שהראנו קודם או להכנס לBook Group ולקבוע קבוצת למידה. בנוסף, הוא יכול לבטל שיעורים וקבוצות, ליצור קשר עם מורה דרך אפליקציית WhatsApp ולשלם על השיעורים.
בתור מורה:
הוא יכול לבחור שעות בהן הוא פנוי ללמד שיעורים, קורסים אותם הוא יכול ללמד, לבקש תשלום מתלמידים וליצור קשר עם התלמיד דרך אפליקציית WhatsApp.

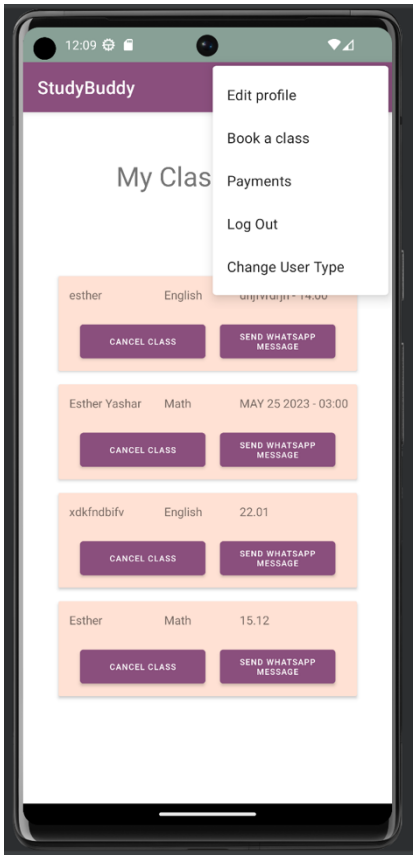
בדיאגרמה זו מתואר כיצד המשתמש מתפעל את המערכת.



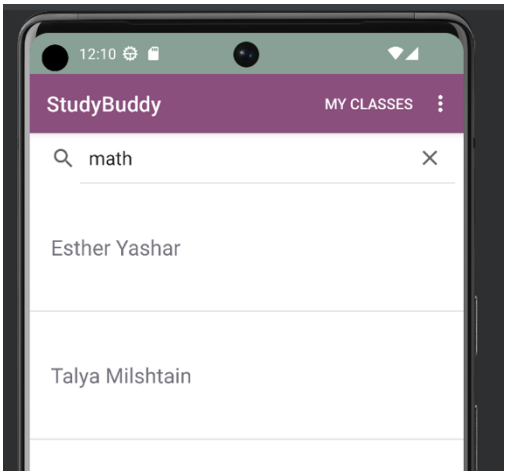
6.2 תמונות מסך



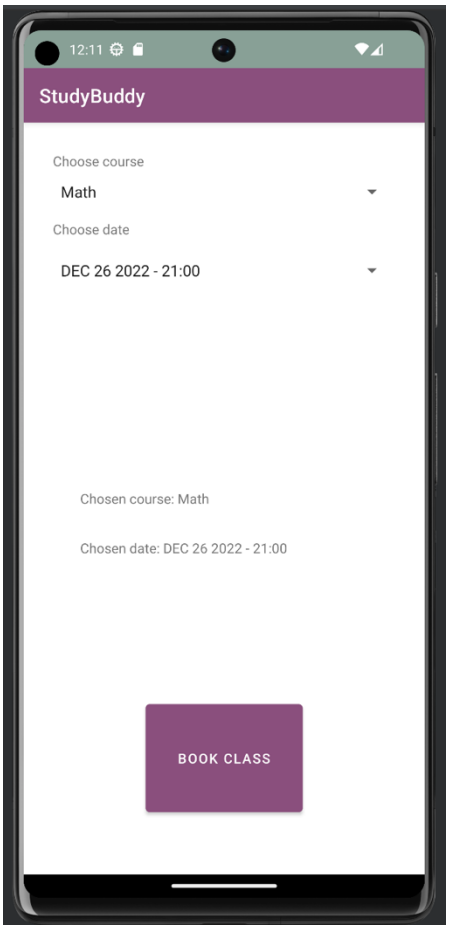
בחירת סוג
משתמש והרשמה



עמוד הבית המראה
אילו שיעורים נקבעו
ועתידים לקרות, וכן
בחירת פעולות נוספות



חיפוש מורים לפי שם קורס



קביעת שיעור לפי תאריך ושעה רלוונטיים

6.3 אובייקטים ופעולות מסך

לאחר לחיצה על כפתור Book Class, וBook Group כלומר לאחר בחירת תאריך ושעה, מורה וקורס על ידי התלמיד, נוצר אובייקט חדש מסוג שיעור השומר את כל המידע הזה, או אובייקט חדש של קבוצה בהתאמה.

7. טבלת דרישות

נתייחס למסמך הדרישות שיצרנו עבור הפרויקט ונראה שמימשנו את הדרישות ואת הצרכים שקבענו:

מס' מזהה	דרישה	מימוש
1	קביעת שיעור	על ידי יצירת האובייקט Class המחזיק את כל המידע על השיעור (תאריך ושעה, מי מלמד ומי תלמיד וכן הלאה)
2	פרסום ימים ושעות	המידע מעודכן על ידי המורה על ידי לחיצה על 'My available times' ומוצג בעת בחירת המורה על ידי התלמיד
3	הרשמה לתלמיד	בעזרת Sign up או Sign in בעת הכניסה לאפליקציה
4	הרשמה למורה	בעזרת Sign up או Sign in בעת הכניסה לאפליקציה
5	עדכון פרטים	על ידי עדכון המידע האישי בעת לחיצה על 'My profile' או 'My courses'
6	מעקב תשלום	כל המידע הרלוונטי נמצא בעמוד 'Payments'
10	עדכון שעות בזמן אמת	ברגע שהמורה מעדכן את השעות אותן הוא מציע, המידע מתעדכן ומופיע אצל התלמיד המעוניין
11	עריכת שינויים ותיקונים במערכת	דרך Android Studio
12	ממשק משתמש נוח והבנה מהירה של איך להשתמש במוצר	האפליקציה נבנתה בתורה שנוחה למשתמש וברורה מאוד. כל פעולה נעשית בעמוד הרלוונטי אליה
13	ביטול ועריכת שיעורים	פעולות אלה נעשה באמצעות לחיצת הכפתור 'Delete class' שנמצא בעמוד הבית של כל משתמש וכן ניתן לערוך שיעור ב'Book Class'

14	סינון מורים לבחירה	הסינון מתבצע בעת חיפוש קורס רלוונטי בעמוד 'Book Class' ולאחר מכן על ידי סינון לפי שעות, תאריכים רלוונטיים
15	פתיחת קבוצה	נעשה על ידי לחיצה על 'Book Group'
16	קביעת קבוצה	נעשה דרך לחיצה על 'Book Group'
17	סינון קבוצה לבחירה	הסינון מתבצע בעת חיפוש קורס רלוונטי בעמוד 'Book Group' ולאחר מכן על ידי סינון לפי שעות, תאריכים רלוונטיים